DT04 Rec'd PCT/PTO 12 OCT 2004

CLAIMS

1. A method of Procedure for the scheduling of a service resource shared among several information packet flows that generate respective associated queues, said flows including synchronous flows ($i = 1, 2, \ldots, N_S$) that require a guaranteed minimum service rate (r_i) and asynchronous flows ($j = 1, 2, \ldots, N_A$) that use the service capacity of said resource that is left unused by the synchronous flows, the method procedure making use of a server (10) and comprising the steps of following operations:

causing makes said server (10) visit the respective queues associated to said flows (i, j) in successive cycles on the basis of the target rotation time value (TTRT), which identifies the time necessary for the server (10) to complete a visit cycle on said respective queues;

associating associates each synchronous flow (i) with a respective synchronous capacity value (H_i) indicating the maximum period of time for which the respective synchronous flow can be serviced before the server moves on;

associating associates each asynchronous flow (j) with a first respective delay value (L_j) that identifies the value that must be made up for the respective queue to have the right to be serviced, and a second respective value (*last_visit_time*) that indicates the instant in which the server (10) visited the respective queue in the previous cycle, determining for said respective queue, the time that has elapsed since the server's previous visit;

servicing services each queue associated to a synchronous flow (i) for a maximum service time relative to said respective value of synchronous capacity (H_i); and

servicing services each queue associated to an asynchronous flow (j) only if the server's visit (10) occurs before the expected instant, said advance being determined as the difference between said target rotation time value (TTRT) and

the time that has elapsed since the server's (10) previous visit and the accumulated delay, so that ~~if positive,~~ this difference defines the maximum service time for each asynchronous queue; and ~~the~~ procedure also includes the operation that defines, at defining said respective synchronous capacity value (H_i) for the queue associated to the i-th synchronous flow by satisfying:

- i) the expressions

$$\sum_{i=1}^{N_S} H_i + \tau_{\max} \leq TTRT$$

$$TTRT \geq \frac{\tau_{\max}}{1-\sum_{h=1}^{N_S} r_h/C}$$

- ii) as well as at least one of the following expressions

$$H_i = \frac{r_i \cdot TTRT}{C} \quad \text{and}$$

$$H_i = \frac{(N_A+\alpha)\cdot r_i/C}{N_A+1-\sum_{h=1}^{N_S} r_h/C}\cdot TTRT$$

where:

- H_i is said respective synchronous capacity value (H_i) for the queue associated to the i-th synchronous flow,
- the summations are extended to all the synchronous flows, equal to N_S,
- N_A is the number of said asynchronous flows,
- τ_{max} is the duration of the longest packet service by said shared service resource,
- TTRT is said target rotation time value,
- C is the service capacity of said shared service resource,
- r_i is the minimum service rate required by the i-th synchronous flow, with $\sum_{h=1}^{N_S} r_h/C < 1$, and
- α is a parameter that gives $\sum_{h=1}^{N_S} r_h/C \leq 1-\alpha$.

2. The method defined in ~~Procedure as per~~ claim 1, wherein ~~characterised by the fact that~~ during each of said successive cycles, said server (10)

performs a double scan on all the queues associated to said synchronous flows ($i = 1, 2, \ldots, N_S$) and then visits the queues associated to said asynchronous flows ($j = 1, 2, \ldots, N_A$).

The method defined in which comprises

3. Procedure as per claim 2, characterised by the fact that it includes the following operations:

- associates to each synchronous flow (i) a further value (Δ_i) indicating the amount of service time that is available to the respective flow,

- during a major cycle of the said double scan, servicing it services each queue associated to a synchronous flow (i) for a period of time equal to the maximum said further value (Δ_i), and

- during a minor cycle of said double scan, servicing it services only one packet of each queue associated to a synchronous flow (i), provided that said further value (Δ_i) is strictly positive.

The method defined in which

4. Procedure as per claim 3, characterised by the fact that it includes the operation of incrementing said further value (Δ_i) by said respective value of the synchronous capacity (H_i) when the queue is visited during the major cycle of said double scan.

The method defined in which

5. Procedure as per claim 3 or claim 4, characterised by the fact that it includes the operation of decrementing said further value (Δ_i) of the transmission time by each packet serviced.

The method defined in wherein

6. Procedure as per any of the claims 3 to 5, characterised by the fact that the service of each queue associated to a synchronous flow (i) during the major cycle of said double scan ends when one of the following conditions occurs:

- the queue is empty,

- the time available, represented by said further value (Δ_i), is not sufficient to service the packet at the front of the queue.

The method defined in which

7. Procedure as per claim 6, characterised by the fact that it includes the operation of resetting said further value (Δ_i) when the respective queue is empty.

The method defined in which

8. Procedure as per any of the claims 3 to 7, characterised by the fact that it includes the operation of decrementing the service time of said further value (Δ_i) in the presence of a service given during the minor cycle of said double scan.

The method defined in wherein

9. Procedure as per any of the claims 3 to 8, characterised by the fact that during said double scan of all the queues associated to said synchronous flows (i), said minor cycle ends when one of the following conditions is satisfied:

- the last queue associated to a synchronous flow (i) has been visited,

- a period of time not less that the sum of the capacities (Hi) of all the queues associated to said synchronous flows (i) has elapsed since the beginning of said major cycle of said double scan.

The method defined in which

10. Procedure as per any of the claims 3 to 9, characterised by the fact that it includes the operation of initialising said further value (Δ_i) to zero.

The method defined in claim 10 wherein

11. Procedure as per any of the previous claims, characterised by the fact that in the case that said difference is negative, each said queue associated to an asynchronous flow (j) is not serviced and the value of said difference is accumulated with said delay (L_i).

The method defined in claim 11 wherein

12. Procedure as per any of the claims 1 to 11, characterised by the fact that the service of a queue associated to an asynchronous flow (j) ends when one of the following conditions is satisfied:

- the queue is empty,
- the time available is not sufficient to transmit the packet that is at the front of the queue.

13. The method defined in claim 12 ~~Procedure as per any of the claims 1 to 12,~~ wherein ~~characterised by the fact that~~ said first respective value (L_j) and said second respective value (*last_visit_time*) are respectively initialised to zero and to the moment of startup of the current cycle when the flow is activated.

14. A ~~System~~ system for the scheduling of a service resource shared among a plurality of ~~several~~ information ~~packets~~ packet flows ~~that~~ generate respective associated queues, ~~S~~said flows including ~~include~~ synchronous flows ($i = 1, 2, \ldots, N_S$) that require a guaranteed minimum service rate and asynchronous flows ($j = 1, 2, \ldots, N_A$) destined to use the service capacity of said resource left unused by the synchronous flows, and comprising ~~The system also includes~~ a server (10) able to visit the respective queues associated to said flows (i, j) in successive cycles, and ~~which~~ is configured to perform the following operations:

- determine a target rotation time value (TTRT) that identifies the time necessary for the server (10) to complete a visiting cycle of said respective queues,
- associate to each synchronous flow (i) a respective synchronous capacity value (H_i) indicating the maximum amount of time for which a synchronous flow can be serviced before moving on to the next,
- associate to each asynchronous flow (j) a first respective delay value (L_j) that identifies the delay that must be made up for the respective queue to have the right to be serviced, and a second respective value (*last_visit_time*) that indicates the instant in which in the previous cycle the server (10) visited the respective queue, determining for said respective queue, the time that has elapsed since the server's (10) previous visit,

service each queue associated to a synchronous flow (i) for a maximum period of time relating to said respective synchronous capacity value (H_i), and

service each queue associated to an asynchronous flow (j) only if the server's visit (10) occurs before the expected instant, said advance being determined as the difference between said target rotation time (TTRT) and the time that has elapsed since the server's (10) previous visit and the accumulated delay; if positive, this difference defines the maximum service time for each said asynchronous queue.

the system being is configured to define said respective synchronous capacity value (H_i) for the queue associated to the i-th synchronous flow so that the following are satisfied:

- i) the expressions

$$\sum_{i=1}^{N_S} H_i + \tau_{\max} \le TTRT$$

$$TTRT \ge \frac{\tau_{\max}}{1 - \sum_{h=1}^{N_S} r_h / C}$$

- ii) as well as at least one of the following expressions

$$H_i = \frac{r_i \cdot TTRT}{C} \quad \text{and}$$

$$H_i = \frac{(N_A + \alpha) \cdot r_i / C}{N_A + 1 - \sum_{h=1}^{N_S} r_h / C} \cdot TTRT$$

where:

- H_i is the said respective synchronous capacity value (H_i) for the queue associated to the i-th synchronous flow,
- the summations are extended to all the synchronous flows, equal to N_S,
- N_A is the number of said asynchronous flows,

- τ_{max} is the service duration of the longest packet by said shared service resource,
- TTRT is said target rotation time value,
- C is the service capacity of said shared service resource,
- r_i is the minimum service rate requested by the i-th synchronous flow, with $\sum_{h=1}^{N_s} r_h/C<1$, and
- α is a parameter that gives $\sum_{h=1}^{N_s} r_h/C \le 1-\alpha$.

15. The ~~S~~ystem as ~~per~~ of claim 14, wherein characterised by the fact that during each of the said successive cycles, said server (10) performs a double scan on all the queues associated to said synchronous flow (i = 1, 2, ..., N_S) and then visits the queues associated to said asynchronous flows (j = 1, 2, ..., N_A).

16. The ~~S~~ystem as ~~per~~ of claim 15, wherein characterised by the fact that:

- a further value (Δ_i), indicating the amount of service time available to the respective flow, is associated to each synchronous flow (i),
- during a major cycle of said double scan, each queue associated to a synchronous flow (i) is serviced for a period of time equal to the maximum further value (Δ_i), and
- during a minor cycle of said double scan the system services only one packet of each queue associated to a synchronised flow (i), provided said further value (Δ_i) is strictly positive.

17. The ~~S~~ystem as ~~per~~ of claim 16, wherein characterised by the fact that said further value (Δ_i) is incremented by said respective synchronous capacity value (H_i) when the queue is visited during the major double scan cycle.

18. The ~~S~~ystem as ~~per~~ of claim 16 or claim 17, wherein characterised by the fact that said further value (Δ_i) is decremented by the transmission time of each packet serviced.

19. The ~~System~~ as per of claim any of the claims 16 to 18, characterised by the fact which is that the system is configured so that the service of each queue associated to a synchronous flow (i) during the major cycle of said double scan ends when one of the following conditions occurs:

- the queue is empty,
- the time available, represented by said further value (Δ_i), is not sufficient to serve the packet at the front of the queue.

20. The ~~System~~ as of per claim 19, wherein characterised by the fact that said further value (Δ_i) is reset when the respective queue is empty.

21. The ~~System~~ as per of claim 20 wherein any of the claims 16 to 20, characterised by the fact that in the presence of a service given during the minor cycle of said double scan, said further value (Δ_i) is decremented by the amount of service time.

22. The ~~System~~ as per of claim any of the claims 16 to 21, wherein characterised by the fact that during said double scan on all the queues associated to said synchronous flows (i), said minor cycle ends when one of the following conditions is satisfied:

- the last queue associated to a synchronous flow (i) has been visited,
- a period of time not less than the sum of the capacities (Hi) of all the queues associated to said synchronous flows (i) has elapsed since the beginning of said major cycle of said double scan.

23. The ~~System~~ as per any of claim of the previous claims 16 to 22, wherein characterised by the fact that said further value (Δ_i) is initialised to zero.

24. The ~~System~~ as per any of claim the previous claims 16 to 23, wherein, characterised by the fact that if said difference is negative, each said queue associated to an asynchronous flow

(j) is not serviced and the value of said difference is accumulated with said delay (L_j).

25. The ~~System~~ as per any ~~of~~ of the claims 14 to 24 wherein characterised by the fact that the service of a queue associated to an asynchronous flow (j) ends when one of the following conditions is satisfied:

- the queue is empty,
- the time available is not sufficient to transmit the packet ~~that~~ the is at the front of the queue.

26. The ~~System~~ as per of claim any of the claims 14 to 25 wherein characterised by the fact that said first respective value (L_j) and said second respective value (*last_visit_time*) are respectively initialised to zero and to the moment of startup of the current cycle when the flow is activated.